

## **BACKGROUND OF THE INVENTION**

### **Field of the Invention**

This invention relates to frame based digital communication systems and more particularly to a method for processing frames having a certain size in systems configured for frames having a different size.

### **Description of the Related Art**

Digital data is often communicated in frames, which are groups of digital data that are processed together, as will be further described below. Often, it is desired to send or receive frames having a certain size in systems configured for different sized frames. For example, in the IS2000.2-A standard for code division multiple access (CDMA) communications, encoded symbols at a rate of  $L$  symbols per frame must be matched with a transmission scheme that processes  $N$  symbols per frame, where  $N$  is greater than  $L$ . In this case, for any one frame, the  $L$  symbols are repeated  $M$  times, where  $M$  is the smallest integer such that  $ML > N$ . Then, the  $ML$  symbols are reduced to  $N$  symbols by deleting (puncturing)  $P$  symbols, where  $P = ML - N$ . There is no appreciable loss of information since in CDMA, many symbols are redundant to provide robust protection against transmission errors; deleting a few redundant symbols will typically not cause any significant problems, especially if the symbols are far apart.

A system designer must decide which symbols to puncture. One conventional choice would be to simply puncture out every  $LM/P$  symbols. Since the symbols are discrete, in practice this means that the nearest integer lower than  $LM/P$  is punctured. For example,  $LM$  is 300 and  $P$  is 100 (i.e. 100 symbols must be deleted), every third symbol may be punctured. Using more realistic numbers for CDMA,  $LM$  may be 2400 and  $P$  864, so that every other symbol is punctured. However, if every other symbol is punctured starting at the beginning of the frame, there will be 672 ( $=2400 - 2 * 864$ ) consecutive symbols at the end of the frame that aren't punctured at all. This non-

uniformity in the puncturing scheme may result in performance degradation since, as mentioned above, it is desirable to keep the distance between deleted symbols as great as possible.

5           An alternate scheme involves puncturing out every  $D1$ th symbol, where  $D1$  is the nearest integer greater than  $LM/P$ , thereby puncturing  $P1$  symbols. In this case, if not enough symbols have been punctured (i.e.  $P1 < P$ ), a second iteration is performed to puncture  $P2 (=P-P1)$  symbols that are  $D1 * \lfloor P1/P2 \rfloor$  symbols apart. (Here, the underline denotes the nearest integer less than the amount in brackets.) Returning to the above  
10   example, in the first pass, every third symbol would be deleted, resulting in the deletion of 800 symbols. After this pass, 64 symbols still must be deleted, which is done by deleting every 36<sup>th</sup> symbol. This second puncturing scheme is more uniform than the first scheme described above. However, this second puncturing scheme requires two iterations, which may be undesirable from an implementation standpoint. Further, this  
15   second scheme may result in two consecutive symbols being punctured, which could result in performance degradation. [Has this been publicly disclosed by Samsung? If not, we need its permission to discuss here.]

          Therefore, it would be desirable to have a more uniform puncturing scheme than  
20   the first scheme without the complications and problems associated with the second scheme.

## SUMMARY OF THE INVENTION

According to the present invention, excess bits are deleted from a frame of data in a single iteration through the data, thereby creating a reduced length frame, wherein the  
5 deleting step is performed such that the distance between any two consecutive deleted bits within any group within a first subset of the frame is A bits, and the distance between any two consecutive deleted bits within any group within a second subset of the frame is B bits, where A is an integer greater than 0 and B is an integer greater than A, and where the first subset and the second subset together form a plurality of consecutive bits.

10

## **BRIEF DESCRIPTION OF THE DRAWINGS**

Other objects and advantages of the invention will become apparent upon reading the following detailed description and upon reference to the accompanying drawings in  
5 which:

Figure 1 is a block diagram of a possible digital communication system that includes a frame matching circuit the implements a frame matching method according to  
10 the present invention.

Figure 2 is a flow chart of a first possible frame matching method according to the present invention.

Figure 3 is a flow chart of a second possible frame matching method according to the present invention.

15 Figure 4 is a flow chart of a second possible frame matching method according to the present invention.

While the invention is susceptible to various modifications and alternative forms, specific embodiments thereof are shown by way of example in the drawings and will  
20 herein be described in detail. It should be understood, however, that the drawings and detailed description thereto are not intended to limit the invention to the particular form disclosed, but on the contrary, the intention is to cover all modifications, equivalents and alternatives falling within the spirit and scope of the present invention as defined by the appended claims.

## **DETAILED DESCRIPTION OF THE INVENTION**

This specification describes frame matching methods in the context of CDMA  
5 systems. However, it will be appreciated that the present invention is not restricted to  
CDMA systems.

As used herein, a “subset” of a frame consists of a least one group of consecutive  
bits within a frame. For example, in a frame with bits ordered 1 through 10, a subset of  
10 the frame may consist of a group of the first two bits of the frame and a group of the last  
three bits of the frame. Another subset of that frame may consist of the third through  
seventh bits of the frame. If reference to more than one subset is made, such subsets are  
assumed not to overlap with one another. The length of a subset or group is the number  
of bits within the subset or group. A subset has a length of at least one (there are no  
15 “null” subsets).

The “distance” between two bits is the number of bits that separate those bits in an  
ordered frame. For example, in a frame with bits ordered 1 through 10, the distance  
between the first bit and the third bit is 1.

20 “Consecutive deleted bits” means a pair of deleted bits that are not separated by  
any other deleted bit.

Figure 1 represents a digital communications system 140 within which the present  
25 invention may be embodied. As shown, the digital system 140 comprises a discrete-time  
channel 142 interposed between an encoder 144 and a decoder 130. Discrete-time  
channel 142 comprises a modulator 146, a channel 148 and a demodulator 150. An  
interleaver 145 is interposed between the encoder 144 and the modulator 146. A  
deinterleaver 151 is interposed between the decoder 130 and the demodulator 150.

Channel 148 may be a transmission channel or a storage medium being written to and read from. Interleaver 145 receives a digital output signal from a rate matching circuit 147, which in turn receives a digital signal from an encoder 144. The interleaver 145 interleaves this digital output signal over a certain time period, which is usually  
5 predetermined and known as a frame. Modulator 146 serves to translate the digital output signal from interleaver 145 into signals suitable for channel 148 and thereafter drives the signals across channel 148.

Channel 148 may suffer from interference that corrupts said signals, the  
10 interference possibly taking form in any combination of additive noise, cross channel interference, multi-path interference, and channel fading. Demodulator 150 serves to receive the signals from channel 148 while minimizing the interference as much as is practical, and thereafter translate the signals into digital signals for input to deinterleaver 151, which deinterleaves the digital signal and provides it to decoder 130. Discrete-time  
15 channel 142 can thus be viewed as a unit accepting digital input signals and producing possibly corrupted digital output signals although the present invention is not limited to noisy channels.

Encoder 144 is a convolutional encoder which serves to add redundancy to input  
20 data signal 152. The output of the encoder 144 is L symbols per frame, which must be matched with the interleaver 145, which interleaves N symbols per frame, where N is greater than L. This matching is performed by the rate matching circuit 147, which repeats the L symbols of an input frame M times, where M is the smallest integer such that  $ML > N$ . Thereafter, the matching circuit 147 punctures P symbols from the repeated  
25 input frame, where  $P = ML - N$ .

According to the present invention, the matching circuit 147 deletes bits in a single iteration through a frame such that the distance that separates consecutive deleted bits is either A or B, where A is an integer greater than 0 and B is an integer greater than

A. Preferably, A is equal to D-1, where D is the greatest integer less than LM/P, where P=LM-N. In this case, B is preferably equal to A+1. For example, if N=1536 bits and L=1200 bits, then M equals 2, D is equal to 2, A is equal to 1 and B is equal to 2. Thus, either 1 or 2 bits separate deleted bits, which results in relatively uniform puncturing.

- 5 The number of deleted bits  $P_{D+1}$  separated by B is preferably equal to LM-PD and the number of deleted bits separated by A ( $P_D$ ) is preferably equal to P- $P_{D+1}$ . As will be described below, there are various implementations of the above described method.

More formally, according to the present invention, LM-N bits are deleted from a  
10 frame of data in a single iteration through the data, thereby creating a reduced length frame, wherein the deleting step is performed such that the distance between any two consecutive deleted bits within any group within a first subset of the frame is A bits, and the distance between any two consecutive deleted bits within any group within a second subset of the frame is B bits, where A is an integer greater than 0 and B is an integer  
15 greater than A, and where the first subset and the second subset together form a plurality of consecutive bits.

Preferably, an entire frame is punctured according to the preferred values of A and B. However, a subset of a frame may be punctured with these (or other) values of A and  
20 B and the remainder of the frame may be punctured by other known methods. For example, the first two bits of a frame may be punctured, and then the remainder of the frame may be punctured with the minimum distance between punctured bits as A or B; the total number of punctured bits should add up to P.

25 As described above, there are many different implementations of the high level method described above. According to one implementation, every D+1th bit is deleted in a frame until  $P_{D+1}$  bits have been deleted. Then, every Dth bit is deleted until the entire frame has been processed. This method will be described in Figure 2. A more uniform scheme involves alternating between deleting every D+1th bit and deleting every Dth bit

until either  $P_{D+1}$  bits separated by B have been deleted or  $P_D$  bits separated by A have been deleted. Thereafter, if the former is true, every Dth bit is deleted, or, if the latter, every  $D+1$ th bit is deleted. This method will be described in Figure 3. According to a third yet still more uniform method, to be described in Figure 4, the following pattern of deletions is repeated: 1 symbol is deleted after skipping A symbols, N symbols separated by B symbols are then deleted, 1 symbol is deleted after skipping A symbols, and then O symbols separated by B symbols are then deleted, where N is the nearest integer less than  $P_{D+1}/P_D$  and O is equal to  $N + 1$ . For example, if A is 1, B is 2, N is 3 and O is 4, then the following bits are deleted 1, 3, 6, 9, 12, 14, 17, 20, 23, 26, 28 ...

Still according to another method, the following pattern of deletions is repeated: N symbols separated by A symbols are deleted and O symbols separated by B symbols are deleted, where N:O is equal to  $P_D : P_{D+1}$ . Preferably, N and O are the smallest integers that effect the ratio N:O. ~~{I threw this in for fun.}~~

Figure 2 is a flow chart that describes the first of the above described implementations. In block 200, a frame of data is received by the matching circuit 147. The frame may be received in any combination of serial and parallel modes. The frame comprises a series of ordered bits, from the first bit to the Lth bit. In block 204, the matching circuit 147 repeats the current frame M times, thereby creating LM bits. In practice, the matching circuit 147 will preferably not store LM bits but will process incoming bits serially, repeating a frame only after first puncturing that frame. For ease of discussion, however, it is assumed that the matching circuit has access to LM bits.

In block 206, the matching circuit 147 sets a counter C equal to P. In block 212, the next B bits of the frame (in this case the first B bits of the frame) are skipped and the next bit is deleted; C is also decremented. In block 214, the matching circuit 147 checks whether C is equal to  $P_D$ . If not, control passes back to block 212. Otherwise, control passes to block 216, where the next A bits are skipped and the next bit deleted; C is



decremented. Control passes to block 218, which checks whether C is 0. If not, control passes back to block 214. Otherwise, the matching circuit is finished.

Figure 3 is a flow chart that describes the second of the above implementations.

5 In block 300, a frame of data is received by the matching circuit 147. In block 310, the matching circuit 147 repeats the current frame M times, thereby creating LM bits. In block 312, the matching circuit 147 sets a counter C equal to 0 and a flag F to 0. In block 314, the next B bits of the frame (in this case the first B bits of the frame) are skipped and the next bit is deleted; C is also incremented. In block 316, the matching  
10 circuit 147 checks whether the flag F has been set (i.e. is equal to 1). If the flag F has been set, control passes to block 318, which checks whether C is equal to P. If so, the routine exits. If not, control passes back to block 314. If the flag F has not been set, control passes to block 320, where the next A bits of the frame are skipped and the next bit is deleted; C is incremented. In block 322, the matching circuit 147 checks whether  
15  $C/2$  is equal to  $P_D$ . If so, the flag F is set in block 323 and control passes to block 318. If not, control passes to block 324, which checks whether  $C/2$  is equal to or greater than  $P_{D+1}$ . If so, control passes back to block 320. Otherwise, control passes back to block 314.

20 Figure 4 is a flow chart that describes the third of the above described implementations. In block 400, a frame of data is received by the matching circuit 147. The frame may be received in any combination of serial and parallel modes. The frame comprises a series of ordered bits, from the first bit to the Lth bit. In block 404, the matching circuit 147 repeats the current frame M times, thereby creating LM bits. In  
25 block 406, the matching circuit 147 sets a counter C equal to P and a flag F to 0. In block 412, the next A bits of the frame (in this case the first A bits of the frame) are skipped and the next bit is deleted; C is also decremented. In block 414, the matching circuit 147 checks whether F is set (equal to 1). If not, control passes to block 416, where a counter CX is set to N, where N is the nearest integer less than  $P_{D+1}/P_D$ . The Flag F is also set to

1. Control passes to block 418, the next B bits are skipped and the next bit deleted; C and CX are decremented. Control passes to block 420, which determines whether CX is 0. If not, control passes back to block 418. If so, control passes back to block 412.

5           In block 414, if the Flag is set to 1, control passes to block 422, where a counter CX is set to O, where  $O=N+1$ . The Flag F is also set to 0. Control passes to block 424, the next B bits are skipped and the next bit deleted; C and CX are decremented. Control passes to block 426, which determines whether CX is 0. If not, control passes back to block 422. If so, control passes to block 428, which determines whether C is equal to 0.  
10   If so, the routine exits. Otherwise, control passes back to block 412.

The third implementation may also be implemented with the following C code, in which Pcnt is the count of the punctured symbols, Pidx(Pcnt) is the index into the unpunctured symbols of the Pcnt-th punctured symbol, for a given Pidx(0)

15   ( $0 < \text{Pidx}(0) < D+1$ ) and  $\text{Pcnt} > 0$ , and a counter RatioCnt is initialized to 0. After the first symbol is punctured, the following steps are repeated until  $\text{Pcnt} > P-2$ :

```
        Pcnt++;
        Pidx(Pcnt) += D;
        RatioCnt +=  $P_{D+1}$ ;
20    If RatioCnt  $\geq P$ ,
            RatioCnt = RatioCnt - P
            Pidx(Pcnt)++
        Endif
```

## 25    Conclusion

Numerous variations and modifications will become apparent to those skilled in the art once the above disclosure is fully appreciated. It is intended that the following claims be interpreted to embrace all such variations and modifications.